MD 2\_2

Dainis Tillers

dt08050

# Kods

/\*\*

\* Dainis Tillers

\* dt08050

\* md2\_2

\*/

#include <stdio.h>

//Globālie mainīgie

int gx; //globālais X

int st; //stāvokli/rezultāts

int test(int x) {

int lx = gx;

gx = gx + 1;

x = x + 1;

if(x == gx && (lx + 1 == gx)) { //pēc vērtības

st = 1;

}

else if(x == gx) { //pēc adreses

st = 2;

}

else if(x == gx+1) { //pēc vērtības un vārda

st = 4;

}

return x;

}

int main()

{

//Inicializē vērtības

gx = 0;

st = 0;

int rez = test(gx);

//Pēc vērtības un rezultāta

if(st == 0 && gx == rez) {

st = 3;

}

//Rezultāta izdrukāšana

switch(st) {

case 1 :

printf("V");

break;

case 2 :

printf("A");

break;

case 3 :

printf("R");

break;

case 4 :

printf("N");

break;

}

return 0;

}

# Padodot pēc vērtības

To, ka parametrs ir padots pēc vērtības ir iespējams noskaidrot, jo izmainot globālo gx vērtību funkcijā, pirms tam piešķirot šo vērtību lokālam funkcijas mainīgajam lx. Izpildot aritmētiskās darbības, galu galā gx un x vērtības sakrīt, kā arī lokālā lx vērtība ir par viens mazāka nekā globālā. Tā kā gx un x ir neatkarīgi, tad tie viens otru negādīgi neietekmē.

# Padodot pēc adreses

Mainot mainīgo gx un x vērtības, tie funkcijas ietvaros ir ar vienādu vērtību. Bet salīdzinot gx ar lokālo lx vērtību šajā gadījumā tā jau atšķirsies par 2 (lx + 2 = gx), jo gan x gan gx izmaiņas ietekmēja gx vērtību.

# Pēc vērtības un vārda

Tā kā x vērtība tiek aprēķināta tikai tad, kad pirmo reizi tai piekļūst, tad funkcijas ietvaros izmainot gx vērtību, pirms vēl x vērtībai kaut reizi ir piekļūts, nozīmē, ka piekļūstot x vērtībai tā saturēs gx + 1. Attiecīgi, kad x vērtībai pieskaita 1 tad tas neietekmē gx un vērtību, un šo vērtību starpība ir 1 (x = gx + 1). Ja padotu tikai pēc vērtības vai adreses, tad gx vērtība sakristu.

# Pēc vērtības un rezultāta

Funkcijas iekšienē nav iespējams atšķir vai parametrs ir padots tikai pēc vērtības vai arī pēc vērtības un rezultāta. Tāpēc izsaucošajā funkcijā tiek nodefinēts mlx mainīgais, kas no sākuma pieņem gx vērtību. Tā kā pēdējā darbība izsauktajā funkcijā ir x – 1, kas nozīmē, ka ja parametrs ir padots pēc vērtības, tad x vērtībai ir jābūt tādai pašai kā sākotnēji padotajai. Funkcijai beidzot savu darbu tas nozīmē, ka arī gx vērtība atgriežoties izsaucošajā funkcijā būs sākotnējā.

# Salīdzinoša tabula

gx = 1; mgx = 1;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | lx = gx | gx = gx + 1 | x = x + 1 | x = x - 1 | main() pēc test() izsaukuma |
| Pēc vērtības | lx = gx = x = 1 | lx = 1; gx = 2; x = 1; | lx = 1; gx = 2; x = 2; | lx = 1; gx = 2; x = 1; | gx = 2; |
| Pēc adreses | lx = gx = x = 1 | lx = 1; gx = x = 2; | lx = 1; gx = x = 3; | lx = 1; gx = x = 2; | gx = 2; |
| Pēc vērtības un vārda | lx = gx = 1 | lx = 1; gx = 2; | lx = 1; gx = 2; x = 3; | lx = 1; gx = x = 3; | gx = 2; |
| Pēc vērtības un rezultāta | lx = gx = x = 1 | lx = 1; gx = 2; x = 1; | lx = 1; gx = 2; x = 2; | lx = 1; gx = x = 2 | gx = 1; |